\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Repetição

**A)** Um açougue que só vende Filet Mignon, Picanha e Alcatra, está com uma superpromoção de carnes embaladas em pacotes de 1kg, com as seguintes tabelas de preços:

|  |  |  |
| --- | --- | --- |
| Carne | Até 10Kg | Acima de 10Kg |
| Filet Mignon | R$ 30,00 por Kg | R$ 15,00 por Kg |
| Picanha | R$ 25,00 por Kg | R$ 12,00 por Kg |
| Alcatra | R$ 30,00 por Kg | R$ 8,00 por Kg |

O açougue aceita 3 formas de pagamento: cartão de débito, dinheiro e ticket refeição. Faça um programa para calcular e exibir o total que um cliente deve pagar por suas compras e o valor a ser devolvido (troco) **ou** a mensagem “**Não há troco**”.

1. Faça uma função que pergunte o nome dos produtos comprados, retornando o valor total da compra do cliente. Como as embalagens são de 1 kg, o nome de um tipo de carne pode aparecer várias vezes, em qualquer ordem. A entrada de itens comprados é finalizada com um nome vazio
2. Faça uma função que receba o valor a ser pago pelo cliente. Esta função deve perguntar o tipo de pagamento: **c** para cartão de débito, **d** para dinheiro ou **t** para ticket. Se o pagamento for realizado por cartão, não há troco. Para os demais tipos de pagamento, pode ou não haver troco. Se for pagamento em dinheiro, será lido o valor pago. Se o pagamento for com tickets, devem ser lidos a quantidade de tickets e o valor de cada ticket (repare que os valores podem ser diferentes).

**Exemplos**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entrada** | | | | | | | **Saída** |
| 120.00 | c | | | | | | Não há troco |
| 120.00 | d | 150.00 | | | | | Troco: 30.00 |
| 120.00 | t | 2 | 80 | 50 | | | Troco: 10.00 |
| 120.00 | t | 4 | 35 | 20 | 30 | 35 | Não há troco |

**Importante**: o seu programa deve processar a despesa de apenas UM cliente.

B) Os índios de uma reserva decidiram cobrar pedágio dos carros que trafegam nas estradas que cruzam suas terras. Quando o motorista chega com o carro onde há uma cancela, deverá fornecer as seguintes informações para que o valor do pedágio seja calculado:

tipo do carro ( Nacional,Importado)

o nº de homens, o nº de mulheres e o nº de crianças que estão no carro

O valor do pedágio por carro é R$ 50,00 mais o seguinte acréscimo por pessoa:

R$ 10,00 por cada homem,

R$ 5,00 por cada mulher e

Para os carros importados, R$ 2,00 por cada criança entre 12 e 18 anos.

Faça um programa que calcule o pedágio a pagar por cada carro ( término da entrada de dados, tipo do carro == 'fim').

Ao final, seu programa deve exibir:

percentual de carros importados

Média de adultos por carro

Obs: A idade das crianças só deve perguntado quando necessário.

C) A Secretaria de Educação mantém as seguintes informações de todas as universidades do estado do Rio de Janeiro:

Nome da universidade

número de cursos oferecidos pela universidade

e para curso

Total de alunos de doutorado

Total de alunos de mestrado

Total de alunos de graduação

Total de bolsas de estudo

Faça um programa que obtenha para cada universidade, os dados de cada um de seus cursos. O programa deve informar:

para cada curso: o percentual de alunos em graduação

a mensagem “No LIMITE” se a quantidade de bolsas de estudos for superior a 60% dos alunos

para cada universidade: O total de faculdades com mais de 6 cursos diferentes

Total de alunos de pós-graduação (mestrado e doutorado)

Número médio de alunos de graduação

D) Num parque de estacionamento de automóveis registra-se, ao longo de um dia, o movimento de entrada e saída de veículos. O parque está aberto diariamente das 8.00 às 24.00 horas.

No registro do estacionamento constam as seguintes informações:

* tipo – (1) horista - (2) diarista
* placa (string 7)
* Horário de entrada: hh/mm
* Horário de saída: hh/mm
* valor pago

A cobrança do estacionamento é diferenciada por turno, sendo que a escolha do turno é determinada pelo horário de entrada:

* manhã: 8:00 – 12:00,
* tarde: 12:01 – 17:00
* noite : 17:01 – 24:00

O cálculo do valor a pagar é dado segundo a seguinte tabela:

|  |  |
| --- | --- |
| 1 ª hora ou fração | Valor básico do turno |
| até 2 horas | 1.5 \* valor básico do turno |
| até 4 horas | 2.5 \* valor básico do turno |
| acima de 4 horas | Valor a pagar por 4 horas + valor a pagar pelos minutos excedentes.  Cálculo do preço do minuto excedente: 0.01% do valor básico do turno |
| acima de 7 horas | valor da diária |

Faça um programa que inicialmente obtenha o valor básico para cada um dos turnos e o valor da diária. A seguir, obtenha o movimento registrado do estacionamento em um dia e confira para cada registro se o valor pago foi correto enviando uma mensagem adequada. Informe, também, o número de diárias pagas e o percentual de valores incorretos e o valor médio pago. O término da entrada de dados é dada pelo tipo 0.

Faça as seguintes funções:

1. Obtem\_Valida\_horario(): Esta função deve pedir ao usuário um horário ( hh/mm). Para validar este horário lido deve-se considerar:

* Se os valores lidos representam um horário de entrada, devem ser superior a 8:00 e inferior 24:00.
* Se os valores lidos representam um horário de saída, devem ser superior ao horário de entrada e inferior ou igual a 24:00.

Esta função deve retornar o horário lido validado.

1. Calcula\_tempo(): esta função recebe o horário de entrada e o horário de saída, calculando o turno e o tempo total (em minutos) que o carro esteve no estacionamento
2. Calcula\_preco(): esta função recebe o valor básico do turno e o tempo total em minutos que o carro esteve no estacionamento e retorna o valor a ser pago.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lista:

A) Faça uma função recursiva que receba uma lista de strings ou de lista de strings e retorne cada uma de suas strings traduzida para a língua do X.

Uma sttring traduzida para a língua do X tem antes de cada vogal a letra X.

Exemplo:

querida --> qxuxerxidxa

maxi --> mxaxxi

prova --> prxovxa

B) Um curso de dança de salão deseja saber se haverá ou não uma nova turma. Para abrir uma turma as seguintes regras devem ser obedecidas:

No mínimo 6 casais (1 homem e 1 mulher)

Nenhum participante pode ter menos de 14 anos

Faça uma função que receba a lista de inscritos onde cada elemento é uma sublista com nome do inscrito, sexo e idade. Esta função deve exibir se é possível formar uma nova turma, mostrando o nome dos casais, ou se não é possível.

**C)** Uma escola promove uma excursão para os alunos formando do ensino médio. A escolha do hotel depende da relação de meninos e meninas inscritos da seguinte forma:

1. Maior quantidade de menina: “Pink Hotel”
2. Maior quantidade de meninos: “Hotel do Sport”
3. Mesma quantidade de meninos e meninas: “Le Hotel”

O gênero dos alunos estão armazenados em uma lista, onde cada elemento é uma sublista com os turnos/turmas dos alunos ou apenas a turma. Por exemplo, a lista abaixo representa as turmas de uma rede escolar onde a primeira escola tem 2 turnos, o primeiro turno tem duas turmas e o segundo turno tem uma turma. A segunda escola só tem uma turma:

LEscolas: [ [ [ [‘f’,’m’],[’f’,’m’,’m’] ], [‘f’,’m’,’f’,’f’ ] ] , [‘f’,’f’,’m’,’m’] ]

1. Faça a função **contaUmaEscola** que receba a lista de alunos de **uma escola** e retorne uma lista com dois elementos, o primeiro é a quantidade de meninas desta escola e o segundo a quantidade de meninos.
2. Faça a função **determinaHotel** que receba a lista da rede Escolar ( lEscolas) e, usando a função **contaUmaEscola**, calcule a quantidade total de meninas e meninos ( em toda a rede) e exiba o nome do hotel escolhido de acordo com a regra descrita anteriormente

D) Sabendo que "meio do intervalo" (mid-range) de uma lista de números é dado pela média do valor máximo(vmax) com o valor mínimo (vmin) armazenados:

mid-range= (vmin + vmax)/2

Escreva uma função que recebe uma lista de reais e calcule o meio do intervalo.

**E)** Um programador resolveu construir um programa para montar o calendário de P2 dos alunos do CB. Este calendário deve ser armazenado em uma lista onde cada elemento possui a seguinte estrutura:

[ [mês,dia], lista de nome de disciplinas cuja avaliação é neste mês/dia]

Exemplo de calendário:

[ [[5,28], [MAT123, QUI123]], # *no dia 28 de maio – disciplinas: MAT123, QUI123*

[[5,29], [CRE123]],

[[6,7], [FIS123, INF123, CAL123]],

[[6,8], [DES123]] ]

Portanto, inicialmente seu programa monta uma lista com as datas das provas de todas as disciplinas do CB. Cada elemento desta lista contém:

[[mês,dia],código da disciplina]

Estes dados são introduzidos via teclado (por enquanto), onde é perguntado no CB:

Código da Disciplina

Data (‘dd/mm’)

Término da entrada de dados: código da disciplina vazio.

Período válido de provas: 15 de maio a 10 de junho

Datas inválidas: enviar mensagem adequada

A seguir, para cada aluno (após fornecer as disciplinas que está inscrito) o programa exibe, na tela, o calendário de suas provas e o dia com mais provas ( bem como o número de provas) no formato descrito abaixo.

Cada aluno deve introduzir, via teclado, sua matrícula, nome, número de disciplinas que está cursando e códigos das disciplinas.

Término da entrada de dados: matrícula == 0

Caso seja digitado uma disciplina que não consta na lista de provas, enviar mensagem.

Exemplo:

|  |  |
| --- | --- |
| Entrada: | Saída na tela |
| Matrícula: 123  Nome: Ana  Nº de disciplinas: 7  Código da disciplina 1: INF123  Código da disciplina 2: MAT123  Código da disciplina 3: DEST123  Código da disciplina 4: CRE123  Código da disciplina 5: QUI123  Código da disciplina 6: FIS123  Código da disciplina 7: CAL123 | Calendário:  28/5: [MAT123, QUI123]  29/5: [CRE123],  7/6: [FIS123, INF123, CAL123]  8/6: [DES123]  Dia com mais provas: 7/6 – com 3 provas |
| Matrícula: 0 |  |

1. Construa a função montaDatasProvas, que monta a lista com as datas das provas de todas as disciplinas do CB como descrito acima. Esta função deve perguntar ao usuário os códigos das disciplinas e as datas (‘dd/mm’), validando-as e retornar a lista construída.

Obs: Término da entrada de dados: código da disciplina vazio.

Período válido de provas: 15 de maio a 10 de junho

Datas inválidas: enviar mensagem adequada

*Após construir e testar sua função montaDatasProvas,* *utilize a função constroiDatasProvas abaixo, para testar as demais funções de seu programa, evitando a redigitação das datas de provas das disciplinas do CB a cada teste.*

def constroiDatasProvas():

L=[[[6,7],'INF123'],[[6,7],'INF124'],

[[5,28],'MAT123'],[[5,29],'CRE123'],

[[5,29],'CRE126'],[[5,28],'QUI123'],

[[6,7],'FIS123'],[[6,7],'CAL123'],

[[6,8],'DES123'],[[6,9],'DES124']

]

return L

1. Construa a função montaCalendário, que recebe a lista com as datas das provas de todas as disciplinas e retorna o calendário deste aluno conforme descrito acima. Esta função deve perguntar ao usuário seus dados ( matr, nome, nº de disciplinas e códigos) e retornar o calendário de suas provas no formato descrito acima.

Dica: inclua novos elementos no final da lista de calendário e antes de retornar, utlize o método sort() que irá ordenar corretamente a lista

Disciplinas sem prova cadastrada: enviar mensagem

*Após construir e testar sua função montaCalendário,* *utilize a função constroiCalendarios abaixo, para testar as demais funções de seu programa, evitando a redigitação das disciplinas que o aluno está cursando*

def constroiCalendario():

L=[ [[5,29], ['CRE123']],

[[6,8], ['DES123']],

[[5,28], ['MAT123', 'QUI123']],

[[6,7], ['FIS123', 'INF123', 'CAL123']]]

L.sort()

return L

1. Construa a função exibeCalendário, que recebe a lista com o calendário de provas do aluno, exibe-o no formato descrito acima e no final informa qual a data com maior número de provas bem como a quantidade de provas nesta data